CLAIMS

1. A computer-implemented method for identifying, in a device space, an effective centerscan object color along an edge between an overscan object and a centerscan object, the overscan object having a higher paint order than the centerscan object, the method comprising:

mapping the edge to the device space;

identifying a set of overscan boundary pixels in the device space, the overscan boundary pixels being device space pixels that are intersected by the edge;

creating a vector pointing in a direction of the centerscan object relative to the edge;

applying the vector to each overscan boundary pixel in the set of overscan boundary pixels to identify a corresponding set of centerscan boundary pixels in the device space; and

mapping each centerscan boundary pixel to the centerscan object to identify a color of the centerscan boundary pixel.

2. The method of claim 1, wherein the centerscan object is a raster image and the overscan object is a vector object.

3. The method of claim 2, wherein an image resolution differs from a device resolution.

4. The method of claim 1, wherein identifying a color of the pixel comprises:

coloring the centerscan boundary pixel in the device space in accordance with a center scan rule.

5. The method of claim 1, wherein creating a vector comprises:

creating a vector specified in device pixels.

6. The method of claim 1, wherein creating a vector comprises:

creating a vector normal to the edge.

7. The method of claim 1, wherein creating a vector comprises:

creating a vector normal to an axis in the device space.

8. The method of claim 1, wherein applying the vector to each overscan object boundary pixel comprises:

identifying a device pixel on the centerscan object side of the edge, adjacent to an overscan boundary pixel, as a centerscan boundary pixel.

9. The method of claim 1, further comprising:

identifying one or more subsections, each subsection including one or more contiguous centerscan boundary pixels having the same color, to be used in trapping

10. A computer program product, residing on a computer-readable medium, for identifying, in a device space, an effective centerscan object color along an edge between an overscan object and a centerscan object, the overscan object having a higher paint order than the centerscan object, the computer program product containing instructions for causing a computer to:

map the edge to the device space;

identify a set of overscan boundary pixels in the device space, the overscan boundary pixels being device space pixels that are intersected by the edge;

create a vector pointing in a direction of the centerscan object relative to the edge;

apply the vector to each overscan boundary pixel in the set of overscan boundary pixels to identify a corresponding set of centerscan boundary pixels in the device space; and

map each centerscan boundary pixel to the centerscan object to identify a color of the centerscan boundary pixel.

11. The computer program product of claim 10, wherein the centerscan object is a raster image and the overscan object is a vector object.

12. The computer program product of claim 11, wherein an image resolution differs from a device resolution.

13. The computer program product of claim 10, wherein the computer program further includes instructions for causing a computer to:

color the centerscan boundary pixel in the device space in accordance with a center scan rule.

14. The computer program product of claim 10, wherein the computer program further includes instructions for causing a computer to:

create a vector specified in device pixels.

15. The computer program product of claim 10, wherein the computer program further includes instructions for causing a computer to:

create a vector normal to the edge.

16. The computer program product of claim 10, wherein the computer program further includes instructions for causing a computer to:

create a vector normal to an axis in the device space.

17. The computer program product of claim 10, wherein the computer program further includes instructions for causing a computer to:

identify a device pixel on the centerscan object side of the edge, adjacent to an overscan boundary pixel, as a centerscan boundary pixel.

18. The computer program product of claim 10, wherein the computer program further includes instructions for causing a computer to:

identifying one or more subsections, each subsection including one or more contiguous centerscan boundary pixels having the same color, to be used in trapping

19. A computer-implemented method for identifying, in a device space, an effective centerscan object color along an edge between an overscan object and a centerscan object,

the centerscan object having a higher paint order than the overscan object, the method comprising:

mapping the edge to the device space;

identifying a set of device space pixels that are intersected by the edge;

determining for each pixel in the set of pixels if a center of the pixel maps to the centerscan object;

identifying the pixel as a centerscan boundary pixel if the center of the pixel maps to the centerscan object;

identifying the pixel as an overscan boundary pixel if the center of the pixel does not map to the centerscan object;

creating a vector pointing in a direction of the centerscan object relative to the edge;

applying the vector to each identified overscan boundary pixel to identify a corresponding centerscan boundary pixel to each identified overscan boundary pixel; and

mapping each centerscan boundary pixel to the centerscan object to identify a color of the centerscan boundary pixel.

20. The method of claim 19, wherein the centerscan object is a raster image and the overscan object is a vector object.

21. The method of claim 20, wherein an image resolution differs from a device resolution.

22. The method of claim 19, wherein identifying a color of the pixel comprises:

coloring the centerscan boundary pixel in the device space in accordance with a center scan rule.

23. The method of claim 19, wherein creating a vector comprises:

creating a vector specified in device pixels.

24. The method of claim 19, wherein creating a vector comprises:

creating a vector normal to the edge.

25. The method of claim 19, wherein creating a vector comprises:

creating a vector normal to an axis in the device space.

26. The method of claim 19, wherein applying the vector to each overscan object boundary pixel comprises:

identifying a device pixel on the centerscan object side of the edge, adjacent to an overscan boundary pixel, as a centerscan boundary pixel.

27 The method of claim 19, further comprising:

identifying one or more subsections, each subsection including one or more contiguous centerscan boundary pixels having the same color, to be used in trapping

28. A computer program product, residing on a computer-readable medium, for identifying, in a device space, an effective centerscan object color along an edge between an overscan object and a centerscan object, the centerscan object having a higher paint order than the overscan object, the computer program product containing instructions for causing a computer to:

map the edge to the device space;

identify a set of device space pixels that are intersected by the edge;

determine for each pixel in the set of pixels if a center of the pixel maps to the centerscan object;

identify the pixel as a centerscan boundary pixel if the center of the pixel maps to the centerscan object;

identify the pixel as an overscan boundary pixel if the center of the pixel does not map to the centerscan object;

create a vector pointing in a direction of the centerscan object relative to the edge;

apply the vector to each identified overscan boundary pixel to identify a corresponding centerscan boundary pixel to each identified overscan boundary pixel; and

map each centerscan boundary pixel to the centerscan object to identify a color of the centerscan boundary pixel.

29. The computer program product of claim 28, wherein the centerscan object is a raster image and the overscan object is a vector object.

30. The computer program product of claim 29, wherein an image resolution differs from a device resolution.

31. The computer program product of claim 28, wherein the computer program further includes instructions for causing a computer to:

color the centerscan boundary pixel in the device space in accordance with a center scan rule.

32. The computer program product of claim 28, wherein the computer program further includes instructions for causing a computer to:

create a vector specified in device pixels.

33. The computer program product of claim 28, wherein the computer program further includes instructions for causing a computer to:

create a vector normal to the edge.

34. The computer program product of claim 28, wherein the computer program further includes instructions for causing a computer to:

create a vector normal to an axis in the device space.

35. The computer program product of claim 28, wherein the computer program further includes instructions for causing a computer to:

identify a device pixel on the centerscan object side of the edge, adjacent to an overscan boundary pixel, as a centerscan boundary pixel.

36. The computer program product of claim 28, wherein the computer program further includes instructions for causing a computer to:

identifying one or more subsections, each subsection including one or more contiguous centerscan boundary pixels having the same color, to be used in trapping.

37. A computer-implemented method for identifying, in a device space, an effective color along one side of an edge between a first centerscan object and a second centerscan object, the method comprising:

mapping the edge to the device space;

identifying a set of device space pixels that are intersected by the edge;

identifying a pixel in the set of device space pixels as a first object boundary pixel if the center of the pixel maps to the first centerscan object;

identifying a pixel in the set of device space pixels as a second object boundary pixel if the center of the pixel does not map to the first centerscan object;

creating a vector pointing in a direction of the second centerscan object relative to the edge;

applying the vector to each identified first object boundary pixel to identify a corresponding second object boundary pixel to each identified first object boundary pixel; and

mapping each second object boundary pixel to the second centerscan object to identify a color of the second object boundary pixel.

38. The method of claim 37, wherein at least one of the first centerscan object and the second centerscan object is a raster image.

39. The method of claim 38, wherein an image resolution differs from a device resolution.

40. The method of claim 37, wherein identifying a color of the pixel comprises:

assigning a color to the second object boundary pixel in the device space in accordance with a centerscan rule.

41. The method of claim 37, wherein creating a vector comprises:

creating a vector specified in device pixels.

42. The method of claim 37, wherein creating a vector comprises:

creating a vector normal to the edge.

43. The method of claim 37, wherein creating a vector comprises:

creating a vector normal to an axis in the device space.

44. The method of claim 37, wherein applying the vector to each first object boundary pixel comprises:

identifying a device pixel on the second object side of the edge, adjacent to a first object boundary pixel, as a second object boundary pixel.

45. The method of claim 37, further comprising:

identifying one or more subsections, each subsection including one or more contiguous centerscan boundary pixels having the same color, to be used in trapping.

46. A computer program product, residing on a computer-readable medium, for identifying, in a device space, an effective color along one side of an edge between a first centerscan object and a second centerscan object, the computer program product containing instructions for causing a computer to:

map the edge to the device space;

identify a set of device space pixels that are intersected by the edge;

identify a pixel in the set of device space pixels as a first object boundary pixel if the center of the pixel maps to the first centerscan object;

identify a pixel in the set of device space pixels as a second object boundary pixel if the center of the pixel does not map to the first centerscan object;

create a vector pointing in a direction of the second centerscan object relative to the edge;

apply the vector to each identified first object boundary pixel to identify a corresponding second object boundary pixel to each identified first object boundary pixel; and

map each second object boundary pixel to the second centerscan object to identify a color of the second object boundary pixel.

47. The computer program product of claim 46, wherein at least one of the first centerscan object and the second centerscan object is a raster image.

48. The computer program product of claim 47, wherein an image resolution differs from a device resolution.

49. The computer program product of claim 46, wherein the computer program further includes instructions for causing a computer to:

assign a color to the second object boundary pixel in the device space in accordance with a centerscan rule.

50. The computer program product of claim 46, wherein the computer program further includes instructions for causing a computer to:

create a vector specified in device pixels.

51. The computer program product of claim 46, wherein the computer program further includes instructions for causing a computer to:

create a vector normal to the edge.

52. The computer program product of claim 46, wherein the computer program further includes instructions for causing a computer to:

create a vector normal to an axis in the device space.

53. The computer program product of claim 46, wherein the computer program further includes instructions for causing a computer to:

identify a device pixel on the second object side of the edge, adjacent to a first object boundary pixel, as a second object boundary pixel.

54. The computer program product of claim 46, wherein the computer program further includes instructions for causing a computer to:

identifying one or more subsections, each subsection including one or more contiguous centerscan boundary pixels having the same color, to be used in trapping